80

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/872,333 | 06/01/2001 | Jose Iborra | CHG-001.3P | 3961 |

26717 7590 01/26/2005

RONALD CRAIG FISH, A LAW CORPORATION
PO BOX 820
LOS GATOS, CA 95032

| EXAMINER | |
|---|---|
| RAMPURIA, SATISH | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2124 | |

DATE MAILED: 01/26/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

|  |  |  |  |
|---|---|---|---|
| *Office Action Summary* | **Application No.**<br>09/872,333 | **Applicant(s)**<br>IBORRA ET AL. |  |
|  | **Examiner**<br>Satish S. Rampuria | **Art Unit**<br>2124 |  |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 03 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *13 October 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-3* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-3* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

## *Response to Amendment*

1. This action is in response to the amendment received on 10/13/2004.
2. The rejection under 35 U.S.C. 112 to claims 1-3 are withdrawn in view of applicant's amendment.
3. Claims 1-3 are amended.

## *Specification*

4. The disclosure is still objected and maintained herein to because of the following informalities:

Specification includes underline words and highlighted words (e.g. pages 8, 9, 40, 56 etc.) It is difficult to read and understand.

Appropriate correction is required.

## *Claim Rejections - 35 USC § 103*

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claim 1-3 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6,199,195 to Goodwin et al., hereinafter called Goodwin, in view of US Patent No. 6,571,232 to Goldberg et al., hereinafter called Goldberg and further in view of US Patent No. 5,742,754 to Tse, hereinafter called Tse.

**Per claim 1:**

Goodwin disclose:

- A process comprising:
- automatically translating a formal language specification written in a formal language defining a full and complete computer program to be automatically written by a computer into a full and complete source code computer program (col. 1, lines 6-10 "automatically generated source code; and more particularly to automatically generated object-oriented source code")that can be compiled into a complete, executable program which can execute by itself on a computer and needs no additional third party source code or source code from existing components or code libraries to be compiled with it to make said complete executable program and which implements the requirements of said formal language specification (col. 6, lines 52-54 "Executable programs (source code objects) are written by the code generator 210 in, for example, platform independent Java programming language"), said specification formal language defining at least classes of objects having attributes, services and relationships with other classes, (col. 1, lines 19-22 "object model describe object classes, attributes of object classes, relationships between object classes and inheritance between object classes"), comprising the following steps:
- using a computer, automatically write computer code that will request user name and password, receive any responses and authenticate the user (col. 9, lines 17-20 "The code generator 330 provides an Application Program Interface (API) that allows other servers to connect, identify a model to be transformed into a server, and invokes code generation

for that server" and col. 16, lines 4-6 "important function is to authenticate client/server programs in order to guarantee security at the software product level");

- using a computer, automatically write computer code that will determine this user's privilege level and query said formal language specification and determine all object attributes this user has privilege to see and query and all services this user can invoke (col. 13, lines 26-28 "The templates… define what is the language of the code… and the unified model determines what is generated… user passes the code generator… a model name… or… Interface Definition Language (IDL) file from model input" and col. 13, lines 52-56 "code generator… support the creation of… IDL, JAVA… based on certain user preferences and selections" also fig. 6 and related discussion);
- using a computer, automatically write computer code which queries said formal language specification for all services of all classes that any authorized user may invoke and identifies an object server which will implement each said service (col. 1, lines 56-58 "automatically map information between an object-oriented application and a structured database, such as a relational database");
- using a computer, automatically write code that will retrieve service arguments for all services from one or more of a user, an object server, and another process, as appropriate (col. 6, lines 37-41 "allow developers to generate objects based on a framework of services they author by composing services based on the object templates into objects that support the composed behaviors and methods");
- using a computer, automatically write code that controls a computer to display means by which and entity can invoke a service, and which receives input to invoke a particular

service and responds by sending a message to the appropriate object server to invoke the service, said message including the necessary arguments for the service to execute (col. 6, lines 37-46 "developers to generate objects based on a framework of services they author by composing services based on the object templates into objects that support the composed behaviors and methods. This is accomplished through the code generator 210, which interfaces with the unified models 206" and col. 9, lines 20-23 "server generated by the code generator 330 is to support Next Generation Information Infrastructure (NGII) services, interfaces, graphs of objects, and data aware objects");

- using a computer, automatically write code that implements an object server for every service, each of which first checks to verify that state transitions are valid and make sense for the current state of objects of which the object service will be altering the state of (col. 6, lines 64-67 to col. 7, lines 1-7 "system also provides a data server (described in reference to FIGS. 3 and 7) for performing run-time object queries that are transformed to access information from enterprise resources with results instantiated between business objects… generated within the composed object service framework… accomplished through the use of an object query service (OMG compliant) and provides a CORBA service to which clients can submit object queries over generated objects and instantiated objects into objects with the composed behaviors defined by the framework");
- using a computer, automatically write code for every object server that verifies preconditions are satisfied before making state transitions of any objects the states of which are acted upon by the object server (col. 7, lines 41-50 "This mapping is manifest in the unified models 206… but also allows specification of how these objects map to

data resources within the enterprise and allow clients to query against these objects, and, at run-time, to have results returned as objects within the complex object framework specified and tailored by a developer");

- using a computer, automatically write code to make all valuation calculations required by said formal language specification of each object server (col. 17, lines 2-5 "run the code generator 330 to generate the source code objects that will support the services in the Interface Definition Language (IDL) file that describes the interface of the objects to the business applications").

Goodwin does not explicitly disclose using a computer, automatically write code to verify that integrity constraints specified in said formal language specification on the values of attributes of objects have been satisfied after execution of a service and take action if said integrity constraints are not satisfied.

However, Goldberg discloses in an analogous computer system using a computer, automatically write code to verify that integrity constraints specified in said specification (col. 3, lines 46-50 "the database schema access object also verifies that an SQL query created by a user is valid by submitting the query to the database engine of the underlying database") on the values of attributes of objects have been satisfied after execution of a service and take action if said integrity constraints are not satisfied (col. 8, lines 22-25 "When the server completes processing of the request, the skeleton code 508 and stub code 506 return the results to the client 500. If an error has occurred, exception information generated by the server or by the ORB is returned").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of code generator to satisfy the attributes in the specification automatically as taught by Goldberg into the method of automatically generating code as taught by Goodwin. The modification would be obvious because of one of ordinary skill in the art would be motivated to fulfill the attributes need as specified in the specification automatically to provide automatically testing of a system as suggested by Goldberg (col. 3, lines 13-24).

Goodwin does not explicitly disclose using a computer, automatically write code for every object server to test trigger relationships specified in said formal language specification after execution of a service and carry out appropriate action if a trigger event has occurred.

However, Tse discloses in an analogous computer system using a computer, automatically write code for every object server to test trigger relationships specified in said specification (col. 1, lines 8-10 “automatically testing software products on a multiplicity of servant computer systems having different hardware configurations” and col. 4, lines 42-43 “test suites are uniquely designed for specific software products”) after execution of a service and carry out appropriate action if a trigger event has occurred (col. 4, lines 43-45 “when test suites are executed during the testing process, errors specific to the software product under test may be uncovered”).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of automatically testing software as taught by Tse into the method of automatically generating code as taught by Goodwin. The modification would be obvious because of one of ordinary skill in the art would be motivated to test software automatically to provide improved testing system with less human interactions as suggested by Tse (col. 2, lines 60-63).

***Claim 2*** is the apparatus claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.

***Claim 3*** is the computer program product claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.

### ***Response to Arguments***

7. Applicant's arguments with respect to claims have been considered but they are not persuasive.

In the remarks, the applicant has argued that:

- Goodwin does not teach or suggest the automatic generation of a complete source code software program to perform user authentication, dynamically determine user privileges as to what information said user is allowed to query and which services said user is allowed to execute, to displays a GUI to invoke services, to collect the arguments of a service (either form a user, another object server or another process) so as to invoke said service, to match requested service against sever objects implementing it, to verifies preconditions are satisfies for every service that produces a state transition, and to perform the calculations dictated by valuation formulas.

- Goldberg does not teach the automatic generation of code of a software system that checks integrity constraints.
- Tse does not teach the automatic generation of code of a software system that checks trigger relationships after execution of a service and carry out appropriate action if a trigger event has occurred.
- The combination of Goodwin, Goldberg, and Tse fails to make a prima facie case because ever if there is no technological incompatibility in combining the teachings of these three prior art patents, the combination fails to teach all the knowledge needed to make the invention of claim 1.

Examiner's response:

- Applicant should note that reference used Goodwin disclose the automatic generation of a complete source code software program which will execute on any complex business framework; see the summary of the invention. Regarding the limitation "perform user authentication" rejection clearly points out the where and why this limitation have been taught by Goodwin. Regarding the limitation "determine user privileges as to what information said user is allowed to query and which services said user is allowed to execute" rejection clearly points out the where and why this limitation have been taught by Goodwin. Applicant has indicated "dynamically" determination of user privilege information which is not found in claim 1, therefore, no patentable weight is given to the limitation "dynamically". Regarding the limitation "to displays a GUI to invoke services, to collect the arguments of a service (either form a user, another object server or another process) so as to invoke said service" rejection clearly points out the where and why this limitation have been taught by Goodwin. Regarding the limitation "to match requested service against sever objects implementing it" rejection clearly points out the where and why this limitation have been taught by Goodwin. Regarding the limitation "to verifies preconditions are satisfies for every service that produces a state transition" rejection clearly points out the where and why this limitation have been taught by Goodwin. Regarding the limitation "to perform the calculations dictated by valuation formulas" rejection clearly points out the where and why this limitation have been taught by Goodwin. Applicant only makes general allegations of improper hindsight reasoning and does not point out any errors in the rejection. Therefore, the rejection is proper and maintained herein.

- Applicant indicated that Goldberg does not teach the automatic generation of code of a software system that checks integrity constraints. However, Goodwin teaches automatic generation of code of a software system and Goldberg teaches checking integrity constraints for the automatic testing system. The rejection clearly points out where Goodwin and Goldberg teach the claimed features and why it would have been obvious to combine their teachings. Applicant only makes general allegations and does not point out any errors in the rejection. Rather, in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Therefore, the rejection is proper and maintained herein.

- Applicant indicated that Tse does not teach automatic generation of code of a software system that checks trigger relationships after execution of a service and carry out appropriate action if a trigger event has occurred. However, Goodwin teaches automatic generation of code of a software system and Tse teaches checks trigger relationships after execution of a service and carry out appropriate action if a trigger event has occurred. The rejection clearly points out where Goodwin and Tse teach the claimed features and why it would have been obvious to combine their teachings. Applicant only makes general allegations and does not point out any errors in the rejection. Rather, in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Therefore, the rejection is proper and maintained herein.
- Regarding the prima facie case, applicant is reminded that the rejection clearly points out where Goodwin, Goldberg, and Tse teach the claimed features and why it would have been obvious to combine their teachings. Applicant only makes general allegations and does not point out any errors in the rejection. Rather, in response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Therefore, the rejection is proper and maintained herein.

*Conclusion*

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Satish S. Rampuria
Patent Examiner
Art Unit 2124
01/24/2005

ANIL KHATRI
PRIMARY EXAMINER